

CERTIFIED COPY OF  
PRIORITY DOCUMENT

11046 U.S. PTO  
09/851405  
05/08/01



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Industrial  
Property Office.

출원 번호 : 특허출원 2000년 제 51651 호  
Application Number

출원 년 월 일 : 2000년 09월 01일  
Date of Application

출원인 : 삼성에스디에스 주식회사  
Applicant(s)



2001      02      05  
년      월      일

특 허 청

COMMISSIONER



|            |                                 |
|------------|---------------------------------|
| 【서류명】      | 특허출원서                           |
| 【권리구분】     | 특허                              |
| 【수신처】      | 특허청장                            |
| 【제출일자】     | 2000.09.01                      |
| 【국제특허분류】   | H04L                            |
| 【발명의 명칭】   | 소프트웨어 테스트 시스템 및 그 방법            |
| 【발명의 영문명칭】 | Software test system and method |
| 【출원인】      |                                 |
| 【명칭】       | 삼성에스디에스 주식회사                    |
| 【출원인코드】    | 1-1998-001790-3                 |
| 【대리인】      |                                 |
| 【성명】       | 이영필                             |
| 【대리인코드】    | 9-1998-000334-6                 |
| 【포괄위임등록번호】 | 1999-025433-5                   |
| 【대리인】      |                                 |
| 【성명】       | 최흥수                             |
| 【대리인코드】    | 9-1998-000657-4                 |
| 【포괄위임등록번호】 | 1999-025556-5                   |
| 【대리인】      |                                 |
| 【성명】       | 이해영                             |
| 【대리인코드】    | 9-1999-000227-4                 |
| 【포괄위임등록번호】 | 2000-002829-9                   |
| 【발명자】      |                                 |
| 【성명의 국문표기】 | 이영석                             |
| 【성명의 영문표기】 | LEE, Young Seok                 |
| 【주민등록번호】   | 650315-1052030                  |
| 【우편번호】     | 136-060                         |
| 【주소】       | 서울특별시 성북구 돈암동 삼성아파트 101동 1408호  |
| 【국적】       | KR                              |
| 【우선권주장】    |                                 |
| 【출원국명】     | KR                              |
| 【출원종류】     | 특허                              |

**【출원번호】** 10-2000-0024608  
**【출원일자】** 2000.05.09  
**【증명서류】** 첨부  
**【심사청구】** 청구  
**【취지】** 특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인  
 이영필 (인) 대리인  
 최홍수 (인) 대리인  
 이해영 (인)  
**【수수료】**  
**【기본출원료】** 20 면 29,000 원  
**【가산출원료】** 23 면 23,000 원  
**【우선권주장료】** 1 건 26,000 원  
**【심사청구료】** 9 항 397,000 원  
**【합계】** 475,000 원  
**【첨부서류】** 1. 요약서·명세서(도면)\_1통

**【요약서】****【요약】**

소프트웨어 테스트 시스템 및 그 방법이 개시된다. 컴퓨터상에서 실행되는 소프트웨어를 테스트하는 본 발명에 따른 소프트웨어 테스트 시스템은 소프트웨어의 오브젝트들을 실행하기 위한 명령어들을 함수화하여 기록한 함수 라이브러리 파일, 소프트웨어의 각 오브젝트를 나타내는 키워드를 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 기록하며, 각 키워드들은 오브젝트 구분자에 의해 구분되는 오브젝트 파일 및 오브젝트 파일로부터 키워드를 순차적으로 읽어들이어 실행하고자 하는 오브젝트를 인식하고, 인식된 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일로부터 호출하여 실행하는 실행 프로그램을 포함을 특징으로 하며, 타겟 소프트웨어의 변경에 따른 유지 및 보수가 매우 간단히 이루어지며, 다른 소프트웨어 및 제품 테스트에 적용시 함수 라이브러리 파일에서 일부 함수를 재정의하는 것만으로도 매우 쉽게 적용할 수 있다.

**【대표도】**

도 3

**【명세서】****【발명의 명칭】**

소프트웨어 테스트 시스템 및 그 방법{Software test system and method}

**【도면의 간단한 설명】**

도 1은 팀테스트를 이용하여 타겟 소프트웨어를 자동 테스트시 스크립트들의 생성을 보이는 도면이다.

도 2는 종래에, 테스트 시나리오대로 타겟 소프트웨어를 실행할 때 생성되는 스크립트 파일들을 나타낸다.

도 3은 본 발명에 따른 소프트웨어 테스트 시스템의 개념을 나타내는 도면이다.

도 4a 내지 도 4c는 도 3에 도시된 오브젝트 파일(14), 실행 프로그램(12) 및 함수 라이브러리 파일(10)의 일 예를 각각 나타내는 도면이다.

도 5는 본 발명에 따른 소프트웨어 테스트 방법의 일 실시예를 설명하기 위한 플로우차트이다.

도 6은 본 발명에 따른 소프트웨어 테스트 시스템의 다른 실시예를 개략적으로 나타내는 블록도이다.

도 7은 사용자 인터페이스(46b)를 통해 오브젝트 데이터 베이스(46a)에 따라 테스트 순서에 따른 키워드 및 인자값의 편집을 할 수 있는 화면을 나타낸다.

도 8은 본 발명에 따른 소프트웨어 테스트 방법의 다른 실시예를 나타내는 흐름도이다.

도 9는 제62단계에서, 스크립트 분석부(48)에 의해 오브젝트 데이터 베이스를 자동으로 구축하는 방법을 나타내는 흐름도이다.

도 10(a)~도 10(e)는 도 9에 도시된 단계별로 오브젝트 데이터 베이스 생성과정을 보이는 도면들이다.

#### 【발명의 상세한 설명】

#### 【발명의 목적】

#### 【발명이 속하는 기술분야 및 그 분야의 종래기술】

- <11> 본 발명은 소프트웨어 테스트에 관한 것으로, 특히, 유지 및 보수가 간편한 소프트웨어 테스트 시스템 및 그 방법에 관한 것이다.
- <12> 일반적으로, 테스트하고자 하는 타겟(target) 소프트웨어를 레코딩 방식에 의해 자동 테스트하는 테스트 툴은 많은 량의 스크립트(script) 파일들을 생성하게된다. 특히, 미국 레이셔널(Rational)사의 팀테스트(TeamTest)와 같이, 선형 스크립트 기술을 이용하는 테스트 툴의 경우, 스크립트 파일은 테스트 케이스 수에 비례하여 생성된다. 예컨대, 팀테스트를 이용하여 타겟 소프트웨어를 테스트하고자하는 경우, 사용자는 먼저 테스트 시나리오를 만들고, 팀테스트를 가동한 상태에서 고안된 테스트 시나리오대로 타겟 소프트웨어를 한 번씩 실행한다. 팀테스트는 테스트 시나리오대로 타겟 소프트웨어를 실행할 때마다 해당 스크립트 파일들을 생성하며, 생성된 스크립트 파일들에 의해 타겟 소프트웨어를 자동적으로 실행하면서 타겟 소프트웨어를 테스트한다.
- <13> 도 1은 팀테스트를 이용하여 타겟 소프트웨어를 자동 테스트시 스크립트들의 생성을 보이는 도면이다.

<14> 도 1을 참조하면, 팀테스트와 같은 선형 스크립트 기술에 의한 테스트 도구는 테스트 시나리오(1)대로 타깃 소프트웨어를 한번씩 실행하여 도구에 의해 생성된 스크립트들(3a1~3an)을 생성한다. 팀테스트는 생성된 스크립트들(3a1~3an)에 의해 타깃 소프트웨어를 실행하면서 타깃 소프트웨어를 자동 테스트한다.

<15> 도 2는 도 1에 도시된 자동 테스트 도구에 의해 생성된 스크립트들(3a1~3an)의 일 예를 나타내는 도면이다. 도 2를 참조하면, 테스트 케이스수 n만큼 스크립트들이 생성되며, 각 스크립트에는 타깃 소프트웨어를 실행하기 위한 명령어(예컨대, 'Window SetContext', 'MenuSelect', 'PushButton')와 데이터('제목없음 - 메모장', '열기', '취소')가 혼재되어 있다.

<16> 한편, 이상에서와 같이 선형 스크립트 기술을 이용한 소프트웨어 테스트는 타깃 소프트웨어가 변경될 경우, 타깃 소프트웨어가 변경된 부분에 해당되는 스크립트들(7)을 수정하거나 부분적으로 재실행하여 스크립트 파일을 다시 생성해야 한다. 만약, 해당 스크립트 파일을 수정하기 힘들거나 부분적으로 새로운 스크립트 파일의 재생성이 불가능하다면, 생성된 스크립트들(3a1~3an)을 모두 폐기하고 테스트 시나리오에 따른 스크립트 파일들을 처음부터 다시 생성해야 한다. 이처럼, 기존의 스크립트 파일들을 폐기하고 새로운 스크립트 파일들을 생성하는데 드는 시간은 다음 수학적 식 1에 의해 측정될 수 있다.

<17> 【수학적 식 1】

스크립트 재생성시간=타깃소프트웨어 변경율×스크립트를 만

드는데 드는 평균시간

<18> ×전체 스크립트수×예수(R)

<19> 여기서, 계수(R)는 프로젝트 규모, 스크립트 사이즈, 전체 스크립트수, 기능수등을 고려하여 부여하여야 하나, 여기서는 1로 정한다. 그리고, 전체 스크립트수를 50개로 한정하고, 타깃 소프트웨어의 변경 화면율을 5%라 하고, 단위 스크립트를 만드는데 드는 시간이 0.3시간이라 할 때, 스크립트 재생성 시간은 0.75시간이 된다. 만약, 타깃 소프트웨어의 변경 규모는 작으나 전체 화면의 50%에 걸쳐 변경이 발생된다면, 즉, 타깃 소프트웨어의 변경 화면율이 50%라면 스크립트 재생성시간은 7.5시간이 된다. 결국, 타깃 소프트웨어의 변경에 따른 소프트웨어 테스트 툴의 유지 및 보수가 복잡하고 많은 시간이 소요됨을 알 수 있다.

<20> 이상에서와 같이, 소프트웨어를 테스트하기 위해 일반적으로 쓰이는 선형 스크립트 기술은 타깃 소프트웨어에 약간의 변경만 발생되어도 스크립트 파일들중 해당 스크립트를 수정하거나 재생성해야 하며, 최악의 경우 전체 스크립트를 다시 생성해야 한다. 또한, 타깃 소프트웨어의 테스트 케이스수에 비례하여 스크립트 파일들이 생성되므로 테스트 케이스가 많을수록 스크립트 파일의 유지 및 보수는 더욱 어려워진다.

<21> 결국, 종래의 선형 스크립트 기술을 이용하여 타깃 소프트웨어를 테스트하면, 테스트 케이스가 많을수록, 타깃 소프트웨어 프로그램 변경이 잦을수록, 소프트웨어 테스트에 걸리는 시간과 노력이 많이 들게된다.

**【발명이 이루고자 하는 기술적 과제】**

<22> 본 발명이 이루고자 하는 기술적 과제는 테스트 대상 소프트웨어의 변경에 따른 유지 보수가 간단한 소프트웨어 테스트 시스템을 제공하는 데 있다.

<23> 본 발명이 이루고자 하는 다른 기술적 과제는 상기 소프트웨어 테스트 시스템에서



수행되는 테스트 방법을 제공하는 데 있다.

- <24> 본 발명이 이루고자 하는 또 다른 기술적 과제는 상기 소프트웨어 테스트 방법을 컴퓨터에서 실행 가능한 프로그램 코드로 기록한 기록 매체를 제공하는 데 있다.

#### 【발명의 구성 및 작용】

- <25> 상기 과제를 이루기 위해, 컴퓨터상에서 실행되는 소프트웨어를 테스트하는 본 발명에 따른 소프트웨어 테스트 시스템은 소프트웨어의 오브젝트들을 실행하기 위한 명령어들을 함수화하여 기록한 함수 라이브러리 파일, 소프트웨어의 각 오브젝트를 나타내는 키워드를 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 기록하며, 각 키워드들은 오브젝트 구분자에 의해 구분되는 오브젝트 파일 및 오브젝트 파일로부터 키워드를 순차적으로 읽어들이어 실행하고자 하는 오브젝트를 인식하고, 인식된 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일로부터 호출하여 실행하는 실행 프로그램을 포함한다.

- <26> 상기 과제를 이루기 위해, 컴퓨터상에서 실행되는 소프트웨어를 테스트하는 본 발명에 따른 테스트 시스템은 상기 소프트웨어의 오브젝트들을 실행하기 위한 명령어들을 함수화하여 기록한 함수 라이브러리 파일, 상기 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 입력되는 상기 소프트웨어의 각 오브젝트에 해당되는 키워드들과 함수 호출에 필요한 인자값들을 저장하는 오브젝트 관리부 및 상기 오브젝트 데이터 베이스를 참조하여 키워드들과 인자값들을 순차적으로 읽어들이고, 키워드에 해당하는 오브젝트를 실행하기 위해 인자값에 대응되는 함수를 함수 라이브러리 파일로부터 호출하여 실행하는 실행 프로그램을 포함한다.

<27>      상기 과제를 이루기 위해, 컴퓨터상에서 실행되는 소프트웨어를 테스트하는 본 발명에 따른 테스트 시스템은 상기 소프트웨어의 오브젝트들을 실행하기 위한 명령어들을 함수화하여 기록한 함수 라이브러리 파일, 최초 테스트 실행시 생성된 스크립트들로부터 테스트 순서에 따른 키워드 및 인자값을 추출하는 함수 분석 및 추출부, 함수 분석 및 추출부로부터 추출되어 순차적으로 입력되는 소프트웨어의 각 오브젝트에 해당되는 키워드들과 함수 호출에 필요한 인자값들을 저장하는 오브젝트 관리부 및 오브젝트 데이터베이스를 참조하여 키워드들과 인자값들을 순차적으로 읽어들이고, 키워드에 해당하는 오브젝트를 실행하기 위해 인자값에 대응되는 함수를 함수 라이브러리 파일로부터 호출하여 실행하는 실행 프로그램을 포함한다.

<28>      상기 다른 과제를 이루기 위해, 컴퓨터상에서 실행되며, 테스트하고자 하는 타깃 소프트웨어를 실행하는 명령어를 함수로 일반화시킨 함수 라이브러리 파일을 구비하는 소프트웨어 테스트 시스템에서, 타깃 소프트웨어를 테스트하는 본 발명에 따른 소프트웨어 테스트 방법은 (a)타깃 소프트웨어의 각 오브젝트를 나타내는 키워드를 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 기록하며, 각 키워드들이 오브젝트 구분자에 의해 구분되는 오브젝트 파일을 생성하는 단계, (b)오브젝트 파일에 기록된 키워드들을 하나씩 순차적으로 읽어들이고, 읽어들이는 키워드에 해당하는 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일로부터 호출하는 단계, (c)(b)단계에서 호출된 함수의 실행에 필요한 소정 개수의 함수 인자들로서 (a)단계에서 읽어들이는 키워드 다음에 연속하는 키워드들을 읽어들이고, (b)단계에서 호출된 함수를 실행하는 단계 및 (d)오브젝트 파일에 실행되지 않은 키워드가 있으면 (b)단계로 진행하여 테스트를 계속 진행하고, 없으면 테스트를 종료하는 단계로 이루어진다.

<29>      상기 다른 과제를 이루기 위해, 컴퓨터상에서 실행되며, 테스트하고자 하는 타깃 소프트웨어의 명령어를 함수로 일반화시킨 함수 라이브러리 파일을 구비하는 소프트웨어 테스트 시스템에서 타깃 소프트웨어를 테스트하는 본 발명에 따른 방법은 타깃 소프트웨어의 각 오브젝트를 나타내는 키워드와 함수 라이브러리 파일로부터 함수 호출을 위한 인자값을 추출하고, 추출된 키워드 및 인자값을 테스트 순서에 따라 저장하여 오브젝트 데이터베이스를 구축하는 (a)단계, 오브젝트 데이터 베이스로부터 키워드 및 인자값을 순차적으로 읽어들이고, 읽어들이는 키워드에 해당하는 오브젝트를 실행하기 위해 인자값에 대응되는 함수를 함수 라이브러리 파일로부터 호출하는 (b)단계, (b)단계에서 호출된 함수의 실행에 필요한 소정 개수의 함수 인자들을 오브젝트 데이터 베이스로부터 연속적으로 읽어들이어 호출된 함수를 실행하는 (c)단계 및 오브젝트 데이터 베이스에 실행되지 않은 키워드가 있으면 (b)단계로 진행하여 테스트를 계속 진행하고, 없으면 테스트를 종료하는 (d)단계로 이루어진다.

<30>      상기 또 다른 과제를 이루기 위해, 컴퓨터상에서 실행되며, 테스트하고자 하는 타깃 소프트웨어의 명령어를 함수로 일반화시킨 함수 라이브러리 파일 및 타깃 소프트웨어의 각 오브젝트를 나타내는 키워드를 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 기록하며, 각 키워드들은 오브젝트 구분자에 의해 구분되는 오브젝트 파일을 구비하는 소프트웨어 테스트 시스템에서 타깃 소프트웨어를 테스트하는 본 발명에 따른 프로그램을 기록한 기록 매체는 (a)오브젝트 파일에 기록된 키워드들을 하나씩 순차적으로 읽어들이고, 읽어들이는 키워드에 해당하는 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일로부터 호출하는 단계, (b)(a)단계에서 호출된 함수의 실행에 필요한 소정 개수의 함수 인자들로서 (a)단계에서 읽어들이는 키워드 다음에 연속하는 키워드들을 읽어

들여, (a)단계에서 호출된 함수를 실행하는 단계 및 (c)오브젝트 파일에 실행되지 않은 키워드가 있으면 (a)단계로 진행하여 테스트를 계속 진행하고, 없으면 테스트를 종료하는 단계로 이루어진다.

<31> 이하, 본 발명에 따른 소프트웨어 테스트 시스템 및 방법을 첨부한 도면들을 참조하여 다음과 같이 설명한다.

<32> 도 3은 본 발명에 따른 소프트웨어 테스트 시스템의 일실시예를 개략적으로 나타내는 도면이다. 본 발명의 일실시예에 따른 소프트웨어 테스트 시스템은 함수 라이브러리 파일(10), 오브젝트 파일(14) 및 실행 프로그램(12)으로 이루어지며, 설명의 편의를 위해 도 2에는 테스트하고자 하는 타겟 소프트웨어(16)를 함께 도시한다. 여기서, 타겟 소프트웨어(16) 및 실행 프로그램(12)은 컴퓨터에서 실행 가능한 프로그램 코드로 이루어진다. 또한, 실행 프로그램(12)은 레이셔널 팀테스트가 제공하는 프로그램 언어 SQABasic을 이용하여 프로그래밍할 수 있다.

<33> 도 3을 참조하여, 함수 라이브러리 파일(10)에는 타겟 소프트웨어(16)의 오브젝트들을 실행하기 위한 명령어들이 함수로 일반화되어 기록된다. 오브젝트 파일(14)에는 타겟 소프트웨어(16)의 오브젝트를 인식을 위한 키워드들이 키워드로서, 타겟 소프트웨어(16)를 테스트하고자 하는 순서에 따라, 순차적으로 기록된다. 실행 프로그램(12)은 오브젝트 파일(14)로부터 키워드를 순차적으로 읽어들이 실행하고자 하는 오브젝트를 인식하고, 인식된 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일로부터 호출하여 함수를 실행한다. 여기서, 함수의 실행은 컴퓨터상에서 타겟 소프트웨어(16)의 테스트 실행을 의미하며, 사용자는 컴퓨터를 통해 테스트 실행 상태를 관찰하며 테스트 실행 결과를 얻을 수 있다.

- <34> 도 4a 내지 도 4c는 도 3에 도시된 오브젝트 파일(14), 실행 프로그램(12) 및 함수 라이브러리 파일(10)의 일 예를 각각 나타내는 도면이다. 도 4a는 오브젝트 파일(14)의 일 예로서, 타깃 소프트웨어의 오브젝트를 인식할 수 있는 오브젝트인식값들 '메뉴선택', '파일(F)→열기(O)', '버튼', '열기', '취소'등이 이 테스트 순서에 따라 순차적으로 기록되며, 각 키워드들은 오브젝트 구분자인 쉼표(,)로 각각 구분된다.
- <35> 도 4b는 실행 프로그램(12)의 일 예로서, 도 4a에 도시된 오브젝트 파일로부터 키워드 '메뉴선택'을 읽어들이어 '메뉴선택'에 해당되는 함수가 'Menu'임을 인식하고, 함수 라이브러리 파일로부터 'Menu' 함수를 호출하여 실행한다.
- <36> 도 4c는 함수 라이브러리 파일의 일 예로서, 명령어 'Window SetContext'와 'PushButton Click'이 함수 'Setf(b)'와 'CancellBut(a)'로서 각각 일반화됨을 보인다.
- <37> 도 5는 본 발명에 따른 소프트웨어 테스트 방법의 일 실시예를 설명하기 위한 플로우차트이다.
- <38> 도 3 내지 도 5를 참조하여, 먼저 테스트하고자 하는 타깃 소프트웨어(16)를 테스트하기에 앞서, 타깃 소프트웨어(16)의 오브젝트들을 실행하기 위한 다수의 명령어들을 함수로 일반화시켜 도 4c에 도시된 바와 같은 함수 라이브러리 파일(10)을 생성한다(제20단계). 제20단계 후에, 키워드으로부터 타깃 소프트웨어(16)의 오브젝트를 인식하고, 인식된 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일(10)로부터 호출하며, 호출된 함수를 실행하여 타깃 소프트웨어의 테스트를 수행하는 도 4b에 도시된 바와 같은 실행 프로그램(12)을 생성한다(제22단계). 제22단계 후에, 타깃 소프트웨어의 오브젝트를 나타내는 키워드를 타깃 소프트웨어를 테스트하고자 하는 테스트 순서에 따라 순차적

으로 기록한 도 4a에 도시된 바와 같은 오브젝트 파일(14)을 생성한다(제24단계).

<39> 제24단계 후에, 실행 프로그램(12)은 오브젝트 파일(14)로부터 키워드를 하나씩 순차적으로 인식한다(제26단계). 이 때, 오브젝트 파일에 기록된 오브젝트들은 도 4a에 도시된 바와 같이 오브젝트 구분자(,)에 의해 각각 분리되어 인식된다. 예컨대, 실행 프로그램(12)은 도 4a에 도시된 오브젝트 파일(14)로부터 '메뉴 선택'이라는 키워드를 읽어들이는 것이다.

<40> 제26단계 후에, 실행 프로그램(12)은 읽어들이는 키워드에 해당하는 함수가 함수 라이브러리 파일(10)에 존재하는가를 검색한다(제28단계). 예컨대, 제26단계에서 실행 프로그램(12)이 '메뉴 선택'이라는 키워드를 읽어들이며, 실행 프로그램(12)은 '메뉴 선택'을 실행하기 위한 함수 'Menu'가 함수 라이브러리 파일에 정의 되어있는가를 검색한다.

<41> 실행 프로그램(12)은 제26단계에서 읽어들이는 오브젝트에 대한 함수가 함수 라이브러리 파일에 정의되어 있으면, 해당 함수를 호출하고(제30단계), 함수 실행에 필요한 함수 인자들로서, 제26단계에서 읽어들이는 키워드 다음에 연속되는 키워드를 읽어들이어 함수 실행을 한다(제32단계). 예컨대, 실행 프로그램은 오브젝트 파일에서 '메뉴선택'에 연속되는 키워드 '파일(F)→열기(O)'를 'Menu'함수를 실행하기 위한 인자로서 읽어들이어 'Menu'함수를 실행한다. 여기서, 함수의 실행은 컴퓨터상에서 타겟 소프트웨어의 테스트의 실행을 의미하며, 컴퓨터상에서 그 실행결과를 확인할 수 있다.

<42> 한편, 제28단계에서, 제26단계에서 읽어들이는 키워드에 해당하는 함수가 함수 라이브러리 파일(10)에 정의되어있지 않다면 테스트를 종료한다.

<43> 제32단계 후에, 실행 프로그램은 오브젝트 파일(14)에 실행되지 않은 키워드들이

남아있는가를 판단하여, 키워드들이 남아있으면 테스트를 계속하기 위해 다음 키워드를 읽어들이며, 실행되지 않은 키워드가 없다면 테스트가 완료된 것으로 판단하고 테스트를 종료한다(제34단계).

<44> 도 6은 본 발명에 따른 소프트웨어 테스트 시스템의 다른 실시예를 개략적으로 나타내는 블록도이다. 본 발명의 다른 실시예에 따른 소프트웨어 테스트 시스템은 함수 라이브러리 파일(40), 오브젝트 관리부(46) 및 실행 프로그램(42)으로 이루어지며, 설명의 편의를 위해 도 6에는 테스트하고자 하는 타겟 소프트웨어(44)를 함께 도시한다. 여기서, 타겟 소프트웨어(44) 및 실행 프로그램(42)은 컴퓨터에서 실행 가능한 프로그램 코드로 이루어진다. 또한, 실행 프로그램(42)은 레이셔널 팀테스트가 제공하는 프로그램 언어 SQABasic을 이용하여 프로그래밍할 수 있다.

<45> 도 6을 참조하여, 함수 라이브러리 파일(40)에는 타겟 소프트웨어(44)의 오브젝트들을 실행하기 위한 명령어들이 함수로 일반화되어 기록된다.

<46> 오브젝트 관리부(42)에는 타겟 소프트웨어(44)의 오브젝트를 실행하기 위한 함수를 호출하는 키워드들과 함수 실행에 필요한 인자들이, 타겟 소프트웨어(44)를 테스트하고자 하는 순서에 따라, 순차적으로 데이터 베이스화되어 저장된다. 구체적으로, 오브젝트 관리부(42)는 사용자 인터페이스(46b) 및 오브젝트 데이터 베이스(46a)를 포함하여 구성된다.

<47> 사용자 인터페이스(46b)는 외부로부터 키워드들 및 인자값들을, 타겟 소프트웨어의 테스트 순서에 따라, 순차적으로 입력하도록 입력 창을 디스플레이한다. 즉, 타겟 소프트웨어를 테스트하고자 사람은 사용자 인터페이스(46b)가 표시하는 입력창을 통해 테스트 순서에 따른 키워드 및 인자값을 직접 입력할 수 있다. 사용자 인터페이스(46b)에 대

한 설명은 도 7을 참조하여 상세히 설명될 것이다.

- <48> 오브젝트 데이터 베이스(46a)는 사용자 인터페이스(46b)로부터 입력되는 키워드들 및 인자값들을 순차적으로 저장한다. 여기서, 오브젝트 데이터 베이스(46a)는 다음 표 1과 같이, 하나의 함수를 실행하는 데 필요한 키워드 및 인자값이 하나의 행에 배열되는 테이블 구조를 갖는다. 예를 들어, '버튼'은 함수를 호출하기 위한 키워드이고, '열기' 및 '취소'는 '버튼'에 의해 호출된 함수를 실행하기 위한 인자값들이다. 또한, 그 다음 행의 '닫기'는 함수를 호출하기 위한 키워드이고, '메모장 정보'는 '닫기'에 의해 호출된 함수를 실행하기 위한 인자값이다. 이처럼, 오브젝트 데이터 베이스(46a)에는 함수를 호출하기 위한 키워드와 호출된 함수를 실행하기 위한 인자값이 하나의 행에 기록되는 테이블 구조를 갖는다.

<49> 【표 1】

|    |        |    |
|----|--------|----|
| 버튼 | 열기     | 취소 |
| 닫기 | 메모장 정보 |    |
|    |        |    |

- <50> 실행 프로그램(42)은 오브젝트 관리부(46)의 오브젝트 데이터 베이스(46a)로부터 행단위로 키워드 및 인자값을 순차적으로 읽어들이 실행하고자 하는 오브젝트를 인식하고, 인식된 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일로부터 호출하여 함수를 실행한다. 여기서, 함수의 실행은 컴퓨터상에서 타깃 소프트웨어(44)의 테스트 실행을 의미하며, 사용자는 컴퓨터를 통해 테스트 실행 상태를 관찰하며 테스트 실행 결과를 얻을 수 있다.

- <51> 한편, 이상에서는 오브젝트 관리부(46)의 오브젝트 데이터 베이스는 테스트하고자



하는 사람이 사용자 인터페이스(46b)를 통해 키워드 및 함수값을 직접 입력함으로써 구축된다. 그러나, 도 6에 도시된 바와 같이 스크립트 분석부(48)를 통해 테스트 순서에 따른 키워드 및 인자값을 자동으로 오브젝트 데이터 베이스(46a)에 저장시킬 수 있다. 구체적으로, 타겟 소프트웨어를 테스트하고자 하는 순서대로 최초 테스트를 실행한다. 그러면, 종래 기술에서 설명된 바와 같이 테스트 스크립트들이 생성된다. 스크립트 분석부(48)는 이처럼 최초 테스트 실행시 생성된 스크립트들로부터 테스트 순서에 따른 키워드 및 인자값을 추출하고, 추출된 키워드 및 인자값을 오브젝트 데이터 베이스(46a)에 저장한다. 스크립트 분석부(48)의 동작에 대해 도 9 내지 도 10을 참조하여 상세히 설명될 것이다.

<52>        도 7은 사용자 인터페이스(46b)를 통해 오브젝트 데이터 베이스(46a)에 따라 테스트 순서에 따른 키워드 및 인자값의 편집을 할 수 있는 화면을 나타낸다. 도 7에 도시된 사용자 인터페이스(46b)는 테스트하고자 하는 사람이 직접 오브젝트 데이터 베이스(46a)를 구축하거나, 또는 타겟 소프트웨어의 프로그램에 변경에 의해 키워드가 변경되는 경우 해당 키워드를 변경시킬 때 이용된다.

<53>        도 7을 참조하여, 테스트하고자 하는 사람은 사용자 인터페이스(46b)를 통해 '키워드'란에 '기동', '화면이동', '조회입력'....등의 키워드 및 인자값을 순차적으로 직접 입력한다. 그리고, 사용자 인터페이스(46b)는 입력된 키워드 및 인자값을 오브젝트 데이터 베이스(46b)에 테스트 순서에 따라 순차적으로 저장한다.

<54>        한편, 전술된 바와 같이 스크립트 분석부(48)를 이용하는 경우, 테스트 순서에 따른 각 키워드 및 인자값들이 자동으로 입력되어 오브젝트 데이터 베이스(46a)에 저장된다.

- <55> 도 8은 본 발명에 따른 소프트웨어 테스트 방법의 다른 실시예를 나타내는 흐름도이다.
- <56> 도 6 및 도 8을 참조하여, 먼저 테스트하고자 하는 타겟 소프트웨어(44)를 테스트 하기에 앞서, 타겟 소프트웨어(44)의 오브젝트들을 실행하기 위한 다수의 명령어들을 함수로 일반화시켜 함수 라이브러리 파일(40, 도 4c 참조)을 생성한다. 이처럼, 함수 라이브러리 파일(40)을 생성한 후, 타겟 소프트웨어의 오브젝트를 나타내며, 함수호출에 필요한 키워드와 호출된 함수의 실행을 위한 인자값을 타겟 소프트웨어를 테스트하고자 하는 테스트 순서에 따라 순차적으로 기록한 오브젝트 데이터 베이스(46a)를 생성한다(제62단계). 오브젝트 데이터 베이스(46a)를 생성하는 방법은 테스트하고자 하는 사람이 사용자 인터페이스(46b)를 통해 키워드 및 인자값을 직접 입력하여 생성하는 방법과, 스크립트 분석부(48)에 의해 자동 생성하는 방법이 있다.
- <57> 제62단계에서 오브젝트 데이터 베이스(46a)가 구축되면, 실행 프로그램(12)은 오브젝트 데이터 베이스(46a)로부터 행단위로 키워드와 인자값을 순차적으로 인식한다(제64단계). 전술된 바와 같이, 실행 프로그램(42)은 오브젝트 데이터 베이스(46a)로부터 행단위로 '버튼', '열기', '취소'의 키워드와 인자값을 읽어들인다. 먼저, 실행 프로그램(42)은 키워드 '버튼'을 인식하여 함수 라이브러리 파일(40)로부터 '버튼'에 해당하는 함수가 존재하는가를 판단한다(제66단계).
- <58> 제66단계에서, '버튼'에 대한 함수가 존재하면 '버튼'에 해당되는 함수를 함수 라이브러리 파일(40)로부터 호출한다(제68단계).
- <59> 제68단계에서 연속되는 키워드 '열기' 및 '취소'를 함수 실행을 위한 인자들로서 호출된 함수에 대응시켜 호출된 함수를 실행한다(제70단계). 여기서, 함수의 실행은 전술

된 바와 같이 컴퓨터상에서 타깃 소프트웨어의 테스트의 실행을 의미하며, 컴퓨터상에서 그 실행결과를 확인할 수 있다.

<60> 제70단계 후에, 실행 프로그램(42)은 오브젝트 데이터 베이스(46a)에 실행할 키워드가 있는가를 판단하여, 실행되지 않은 키워드가 남아있으면 행단위의 키워드들을 읽어 들여 함수 실행을 하며, 실행되지 않은 키워드가 없다면 테스트가 완료된 것으로 판단하고 테스트를 종료한다(제72단계).

<61> 도 9는 제62단계에서, 스크립트 분석부(48)에 의해 오브젝트 데이터 베이스를 자동으로 구축하는 방법을 나타내는 흐름도이다.

<62> 도 10(a)~도 10(e)는 도 9에 도시된 단계별로 오브젝트 데이터 베이스 생성과정을 보이는 도면들이다.

<63> 이제, 도 6, 도 9 및 도 10을 참조하여, 오브젝트 데이터 베이스 자동 구축방법을 상세히 설명한다.

<64> 스크립트 분석부(48)을 이용하여 오브젝트 데이터 베이스를 자동으로 구축하기 위해서는, 먼저 타깃 소프트웨어(44)를 테스트하고자 하는 순서에 따라 실행하여 테스트 실행 스크립트들을 생성한다. 도 10(a)에는 메모장에 대한 간단한 테스트를 실행한 결과 생성되는 테스트 실행 스크립트이다.

<65> 이처럼, 테스트 실행 스크립트가 생성되면 도 10(b)에 도시된 바와 같이, 테스트 실행 스크립트들의 모든 단어를 순차적으로 배열하여 저장한다(제100단계). 이 때, 테스트 실행 스크립트들의 각 단어들을 저장하고 있는 배열 각각에는 액세스할 수 있는 어드레스가 할당되어 있다.

<66> 제100단계 후에, 저장된 배열을 순차적으로 검색하여(제105단계) 기 정의된 함수를 특징짓는 신택스(syntax)가 존재하는가를 판단한다(제110단계). 예를 들어, 함수 라이 브러리 파일(40)에 기 정의된 함수중에 다음과 같은 함수가 있다고 가정한다.

<67> Function CancelBut(a,b)

<68> Window SetContext, 'Caption='+a+", "

<69> PushButton Click, 'Text='+b+"

<70> End Function

<71> 위의 함수를 기동해서 테스트하기 위해서는 'CancelBut'이라는 함수를 호출할 수 있어야 하고, "Caption="과 "Text=" 다음에는 각각 캡션값과 텍스트값이 있어야 한다. 이 때, 도 10(b)에 도시된 배열을 참조하면, 'PushButton'이라는 단어 전후 일정한 거리에 필요한 캡션값과 텍스트값이 존재함을 알 수 있으며, 'PushButton'을 신택스 단어로 한다

<72> 스크립트 분석부(48)는 도 10(b)에 도시된 배열을 순차적으로 검색하여 'PushButton'이라는 신택스 단어가 들어 있는지 비교한다. 예컨대, 'PushButton'의 위치가 100번지라면 캡션값의 위치는 98번지이며 텍스트값은 102번지이며 그 값은 각각 '메모장'과 '아니오'가 될 것이다.

<73> 이 때, 실행 프로그램(42)에 정해진 대로 함수호출에 필요한 키워드 '버튼'을 부여하며, 함수 실행을 위한 인자값 '메모장' 및 '아니오'와 함께 도 10(c)에 도시된 바와 같이 하나의 행으로 새롭게 배열하여 저장하며,

<74> 계속해서, 윈도우의 닫기 기능을 수행하는 함수 ExitWin이 아래와 같이 정의 되어

있고 이것을 호출하는 키워드가 '닫기'라고 가정한다.

<75>       Function ExitWin(a)

<76>       Window SetContext,'Caption='+a+", "

<77>       Window CloseWin, ", "

<78>       위의 함수를 기동해서 테스트하기 위해서는 'ExitWin'이라는 함수를 호출할 수 있어야 하고, "Caption=" 다음에는 캡션값이 있어야 한다. 도 10(b)에 도시된 배열을 참조하면, 'CloseWin'이라는 단어 전 일정한 거리에 필요한 캡션값이 존재함을 알 수 있다.

<79>       스크립트 분석부(48)는 도 10(b)에 도시된 배열을 순차적으로 검색하여 'CloseWin'이라는 단어가 들어 있는지 비교한다. 예컨대, 'CloseWin'의 위치가 50번지라면 캡션값의 위치는 48번지이며 그 값은 '메모장 정보'가 될 것이다.

<80>       이 때, 실행 프로그램(42)에 정해진 대로 함수호출에 필요한 키워드 '닫기'를 부여하며, 함수 실행을 위한 인자값 '메모장 정보'와 함께 도 10(d)에 도시된 바와 같이 하나의 행으로 새롭게 배열하여 저장한다.

<81>       이상에서와 같이, 제100단계에서 저장된 배열에 기 정의된 함수를 특징짓는 신택스가 존재한다고 판단되면, 신택스에 해당되는 단어의 앞, 뒤 배열들을 탐색하여 소정 거리내에 위치하는 함수실행을 위한 인자값을 추출한다(제115단계).

<82>       제115단계 후에, 실행 프로그램(42)을 참조하여, 신택스에 해당되는 함수 호출을 위한 키워드를 부여하고, 신택스에 해당되는 단어의 어드레스 정보와 함께 키워드, 인자값을 행단위로 새롭게 배열하여 저장한다(제120단계).

<83>       한편, 도 10(d)를 참조하면, 키워드 '버튼'에 해당되는 신택스의 어드레스는 100이

고, 키워드 '닫기'에 해당되는 신택스의 어드레스는 50이다. 만약, 이 순서대로 오브젝트 데이터 베이스에 저장을 한다면, 실행 프로그램(42)이 오브젝트 데이터 베이스(46)를 참조하여 테스트 수행시 '버튼'에 대한 함수가 먼저 실행된 후 '닫기'에 대한 함수가 실행된다. 그러나, 원래의 테스트 순서는 '닫기'에 대한 함수 실행 후, '버튼'에 대한 함수가 실행되는 것이다. 따라서, 도 10(e)에 도시된 바와 같이, 신택스의 어드레스에 따라 그 순서를 다시 정렬한 후, 오브젝트 데이터 베이스(46)에 저장한다(제125단계).

<84> 도 11은 본 발명의 일 실시예에 따른 소프트웨어 테스트 방법과 종래 기술에 따른 소프트웨어 테스트 방법을 적용하여 타겟 소프트웨어를 테스트한 결과 작업시간을 비교한 결과를 보이는 도면이다.

<85> 도 11을 참조하면, 타겟 소프트웨어를 테스트하고자하는 테스트항목이 작은 경우 종래 기술에 따른 소프트웨어 테스트 방법이 유리하다. 그 이유는 오브젝트 데이터 베이스를 구축하는 데 걸리는 시간 때문이다. 그러나, 테스트 항목이 많아질수록 종래에는 타겟 소프트웨어의 변경에 따른 스크립트 파일을 수정하는 데 걸리는 시간이 많아진다. 반면, 본 발명의 다른 실시예에 따르면 타겟 소프트웨어의 테스트 항목이 많아지더라도 테스트에 걸리는 시간은 크게 늘어나지 않음을 알 수 있다.

<86> 이상에서와 같이, 본 발명에 따른 소프트웨어 테스트 시스템 및 방법에 따르면, 관리대상이 되는 스크립트 파일이 없으므로, 타겟 소프트웨어의 변경에 따라 스크립트 파일을 고치는데 따르는 유지 및 보수 시간이 없다. 즉, 타겟 소프트웨어의 변경이 발생하여도 키워드가 변경되지 않았다면 오브젝트 파일을 변경할 필요없이 그대로 사용하면 된다. 따라서, 타겟 소프트웨어가 변경되어도 함수 라이브러리 파일, 실행 프로그램 또는

오브젝트 파일을 변경할 필요가 없으므로, 타깃 소프트웨어의 변경에 따른 유지 및 보수 공수가 발생되지 않는다.

<87> 또한, 타깃 소프트웨어에 중대한 변경이 발생하여 키워드가 변경되더라도 키워드를 기록한 오브젝트 파일을 고치는데 드는 공수만이 필요하며, 오브젝트 파일을 수정하는데 드는 공수는 0.5시간 이내이다.

<88> 결국, 타깃 소프트웨어의 변경에 맞추어 테스트 시스템의 유지 및 보수작업은 0.5시간 이내에 이루어질 수 있다. 종래에는 스크립트수, 소프트웨어의 변경율, 프로젝트 규모등 많은 변수가 소프트웨어 테스트 시스템의 유지 및 보수에 드는 공수에 영향을 미친다. 그러나, 본 발명에 따른 소프트웨어 테스트 시스템은 타깃 소프트웨어에 중대한 설계 변경이 발생한 경우에만 유지 및 보수 공수가 들며, 이 때에도 오브젝트 파일을 수정하는 정도로 유지 및 보수는 매우 간단히 이루어진다. 또한, 본 발명에 따른 소프트웨어 테스트 시스템 및 방법을 다른 소프트웨어 및 제품 테스트에 적용시 함수 라이브러리 파일에서 일부 함수를 재정의하는 것만으로 쉽게 적용할 수 있다.

<89> 본 발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플라피디스크, 광데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

**【발명의 효과】**

<90> 상술한 바와 같이, 본 발명에 따른 소프트웨어 테스트 시스템 및 방법은 타겟 소프트웨어의 변경에 따른 유지 및 보수가 매우 간단히 이루어지며, 다른 소프트웨어 및 제품 테스트에 적용시 함수 라이브러리 파일에서 일부 함수를 재정의하는 것으로 쉽게 적용할 수 있다.



**【특허청구범위】****【청구항 1】**

컴퓨터상에서 실행되는 소프트웨어를 테스트하는 테스트 시스템에 있어서,

상기 소프트웨어의 오브젝트들을 실행하기 위한 명령어들을 함수화하여 기록한 함수 라이브러리 파일;

상기 소프트웨어의 각 오브젝트를 나타내는 키워드를 상기 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 기록하며, 각 키워드들은 오브젝트 구분자에 의해 구분되는 오브젝트 파일; 및

상기 오브젝트 파일로부터 키워드를 순차적으로 읽어들이어 실행하고자 하는 오브젝트를 인식하고, 인식된 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일로부터 호출하여 실행하는 실행 프로그램을 포함하는 것을 특징으로 하는 소프트웨어 테스트 시스템.

**【청구항 2】**

컴퓨터상에서 실행되며, 테스트하고자 하는 타겟 소프트웨어의 명령어를 함수로 일반화시킨 함수 라이브러리 파일을 구비하는 소프트웨어 테스트 시스템에서 상기 타겟 소프트웨어를 테스트하는 방법에 있어서,

(a) 상기 타겟 소프트웨어의 각 오브젝트를 나타내는 키워드를 상기 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 기록하며, 각 키워드들은 오브젝트 구분자에 의해 구분되는 오브젝트 파일을 생성하는 단계;

(b)상기 오브젝트 파일에 기록된 키워드들을 하나씩 순차적으로 읽어들이고, 읽어

들인 키워드에 해당하는 오브젝트를 실행하기 위한 함수를 상기 함수 라이브러리 파일로부터 호출하는 단계;

(c) 상기 (b)단계에서 호출된 함수의 실행에 필요한 소정 개수의 함수 인자들로서 상기 (a)단계에서 읽어들이는 키워드 다음에 연속하는 키워드들을 읽어들이며, 상기 (b)단계에서 호출된 함수를 실행하는 단계; 및

(d)상기 오브젝트 파일에 실행되지 않은 키워드가 있으면 상기 (b)단계로 진행하여 테스트를 계속 진행하고, 없으면 테스트를 종료하는 단계로 이루어지는 것을 특징으로 하는 소프트웨어 테스트 방법.

### 【청구항 3】

컴퓨터상에서 실행되며, 테스트하고자 하는 타겟 소프트웨어의 명령어를 함수로 일변화시킨 함수 라이브러리 파일 및 상기 타겟 소프트웨어의 각 오브젝트를 나타내는 키워드를 상기 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 기록하며, 각 키워드들은 오브젝트 구분자에 의해 구분되는 오브젝트 파일을 구비하는 소프트웨어 테스트 시스템에서 상기 타겟 소프트웨어를 테스트하는 프로그램을 기록한 기록 매체에 있어서,

(a) 상기 오브젝트 파일에 기록된 키워드들을 하나씩 순차적으로 읽어들이고, 읽어들이는 키워드에 해당하는 오브젝트를 실행하기 위한 함수를 상기 함수 라이브러리 파일로부터 호출하는 단계;

(b)상기 (a)단계에서 호출된 함수의 실행에 필요한 소정 개수의 함수 인자들로서 상기 (a)단계에서 읽어들이는 키워드 다음에 연속하는 키워드들을 읽어들이며, 상기 (a)단계

에서 호출된 함수를 실행하는 단계; 및

(c)상기 오브젝트 파일에 실행되지 않은 키워드가 있으면 상기 (a)단계로 진행하여 테스트를 계속 진행하고, 없으면 테스트를 종료하는 단계로 이루어지는 것을 특징으로 하는 소프트웨어 테스트 프로그램을 컴퓨터에서 실행 가능한 프로그램코드로 기록한 기록 매체.

#### 【청구항 4】

컴퓨터상에서 실행되는 소프트웨어를 테스트하는 테스트 시스템에 있어서,

상기 소프트웨어의 오브젝트들을 실행하기 위한 명령어들을 함수화하여 기록한 함수 라이브러리 파일;

상기 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 입력되는 상기 소프트웨어의 각 오브젝트에 해당되는 키워드들과 함수 호출에 필요한 인자값들을 저장하는 오브젝트 관리부; 및

상기 오브젝트 데이터 베이스를 참조하여 키워드들과 인자값들을 순차적으로 읽어들이고, 키워드에 해당하는 오브젝트를 실행하기 위해 인자값에 대응되는 함수를 함수 라이브러리 파일로부터 호출하여 실행하는 실행 프로그램을 포함하는 것을 특징으로 하는 소프트웨어 테스트 시스템.

#### 【청구항 5】

제4항에 있어서, 상기 오브젝트 관리부는

외부로부터 상기 키워드들 및 상기 인자값들을 순차적으로 입력하도록 입력 창을 디스플레이하는 사용자 인터페이스; 및

상기 사용자 인터페이스로부터 입력되는 키워드들 및 인자값들을 순차적으로 저장하는 오브젝트 데이터 베이스를 포함하는 것을 특징으로 하는 소프트웨어 자동 테스트 시스템.

#### 【청구항 6】

컴퓨터상에서 실행되는 소프트웨어를 테스트하는 테스트 시스템에 있어서,

상기 소프트웨어의 오브젝트들을 실행하기 위한 명령어들을 함수화하여 기록한 함수 라이브러리 파일;

최초 테스트 실행시 생성된 스크립트들로부터 테스트 순서에 따른 키워드 및 인자값을 추출하는 함수 분석 및 추출부;

상기 함수 분석 및 추출부로부터 추출되어 순차적으로 입력되는 상기 소프트웨어의 각 오브젝트에 해당되는 키워드들과 함수 호출에 필요한 인자값들을 저장하는 오브젝트 관리부; 및

상기 오브젝트 데이터 베이스를 참조하여 키워드들과 인자값들을 순차적으로 읽어들이고, 키워드에 해당하는 오브젝트를 실행하기 위해 인자값에 대응되는 함수를 함수 라이브러리 파일로부터 호출하여 실행하는 실행 프로그램을 포함하는 것을 특징으로 하는 소프트웨어 테스트 시스템.

#### 【청구항 7】

컴퓨터상에서 실행되며, 테스트하고자 하는 타겟 소프트웨어의 명령어를 함수로 일반화시킨 함수 라이브러리 파일을 구비하는 소프트웨어 테스트 시스템에서 상기 타겟 소프트웨어를 테스트하는 방법에 있어서,

(a) 상기 타깃 소프트웨어의 각 오브젝트를 나타내는 키워드와 상기 함수 라이브러리 파일로부터 함수 호출을 위한 인자값을 추출하고, 추출된 키워드 및 인자값을 테스트 순서에 따라 저장하여 오브젝트 데이터베이스를 구축하는 단계;

(b)상기 오브젝트 데이터 베이스로부터 키워드 및 인자값을 순차적으로 읽어들이고, 읽어들이는 키워드에 해당하는 오브젝트를 실행하기 위해 인자값에 대응되는 함수를 상기 함수 라이브러리 파일로부터 호출하는 단계;

(c) 상기 (b)단계에서 호출된 함수의 실행에 필요한 소정 개수의 함수 인자들을 상기 오브젝트 데이터 베이스로부터 연속적으로 읽어들이어 호출된 함수를 실행하는 단계; 및

(d)상기 오브젝트 데이터 베이스에 실행되지 않은 키워드가 있으면 상기 (b)단계로 진행하여 테스트를 계속 진행하고, 없으면 테스트를 종료하는 단계로 이루어지는 것을 특징으로 하는 소프트웨어 테스트 방법.

#### 【청구항 8】

제7항에 있어서, 상기 (a)단계는

(a1) 상기 타깃 소프트웨어를 테스트하고자 하는 순서에 따라 실행하여 테스트 실행 스크립트들을 생성하는 단계;

(a2)생성된 테스트 실행 스크립트들의 모든 단어를 배열하여 저장하고, 각 배열에 액세스 가능한 어드레스를 부여하는 단계;

(a3) 상기 (a2)단계에서 저장된 배열을 순차적으로 검색하여 기 정의된 함수를 특징짓는 신택스가 존재하는가를 판단하여, 신택스가 존재하지 않으면 종료하는 단계;

(a4)상기 (a3)단계에서 선택스에 해당되는 단어가 존재하면 그 단어의 앞, 뒤 배열들을 탐색하여 상기 추출된 키워드와 소정 거리내에 위치하는 인자값을 추출하고, 해당 함수의 호출에 필요한 키워드를 부여하는 단계;

(a5) 상기 키워드, 인자값 및 상기 선택스에 해당되는 단어의 어드레스와 함께 행단위로 임시 저장하는 단계;

(a6)상기 (a5)단계에서 저장된 행단위의 키워드 및 인자값을 상기 선택스 단어의 어드레스에 따라 정렬하는 단계; 및

(a6)상기 (a5)단계에서 정렬된 순서로 오브젝트 데이터 베이스에 저장하는 단계로 이루어지는 것을 특징으로 하는 소프트웨어 테스트 방법.

#### 【청구항 9】

컴퓨터상에서 실행되며, 테스트하고자 하는 타겟 소프트웨어의 명령어를 함수로 일반화시킨 함수 라이브러리 파일과, 상기 타겟 소프트웨어를 테스트하고자 하는 순서에 따른 키워드 및 인자값을 행단위로 저장하고 있는 오브젝트 데이터 베이스를 구비하는 소프트웨어 테스트 시스템에서 상기 오브젝트 데이터 베이스를 자동으로 구축하는 프로그램을 기록한 기록 매체에 있어서,

(a1) 상기 타겟 소프트웨어를 테스트하고자 하는 순서에 따라 실행하여 테스트 실행 스크립트들을 생성하는 단계;

(a2)생성된 테스트 실행 스크립트들의 모든 단어를 배열하여 저장하고, 각 배열에 액세스 가능한 어드레스를 부여하는 단계;

(a3) 상기 (a2)단계에서 저장된 배열을 순차적으로 검색하여 기 정의된 함수를 특징짓는 신택스가 존재하는가를 판단하여, 신택스가 존재하지 않으면 종료하는 단계;

(a4)상기 (a3)단계에서 신택스에 해당되는 단어가 존재하면 그 단어의 앞, 뒤 배열들을 탐색하여 상기 추출된 키워드와 소정 거리내에 위치하는 인자값을 추출하고, 해당 함수의 호출에 필요한 키워드를 부여하는 단계;

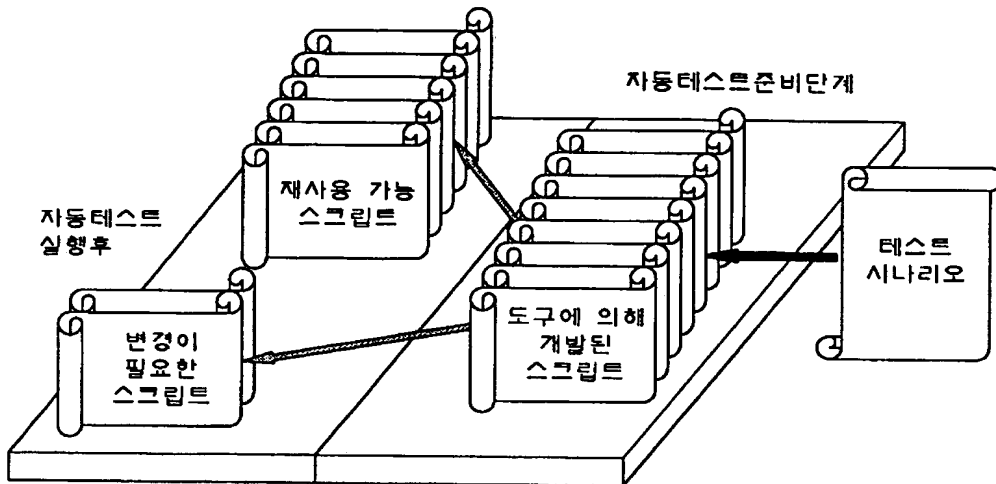
(a5) 상기 키워드, 인자값 및 상기 신택스에 해당되는 단어의 어드레스와 함께 행단위로 임시 저장하는 단계;

(a6)상기 (a5)단계에서 저장된 행단위의 키워드 및 인자값을 상기 신택스 단어의 어드레스에 따라 정렬하는 단계; 및

(a6)상기 (a5)단계에서 정렬된 순서로 오브젝트 데이터 베이스에 저장하는 단계로 이루어지는 것을 특징으로 하는 오브젝트 데이터 베이스 자동 구축 프로그램을 컴퓨터에서 실행 가능한 프로그램 코드로 기록한 기록 매체.

## 【도면】

【도 1】

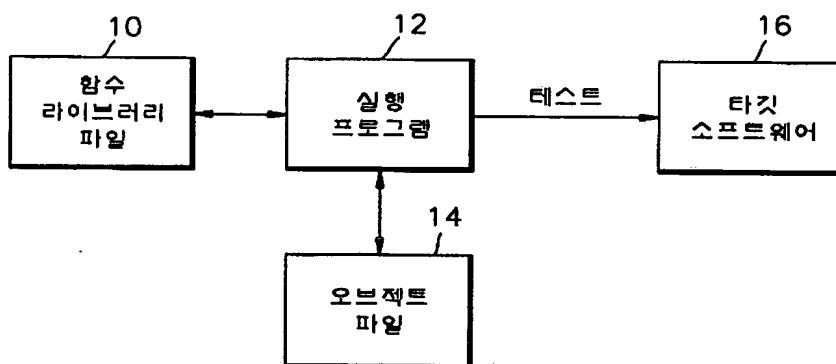


【도 2】

```

1 Window SetContext, "Caption=제목 없음 -메모장", ""
  MenuSelect "파일(F)->열기(O)..."
:
n Window SetContext, "Caption=열기", ""
  PushButton Click, "Text=취소"
  
```

【도 3】



【도 4a】

메뉴 선택, 파일(F) → 열기(O)..., 버튼, 열기, 취소, .....



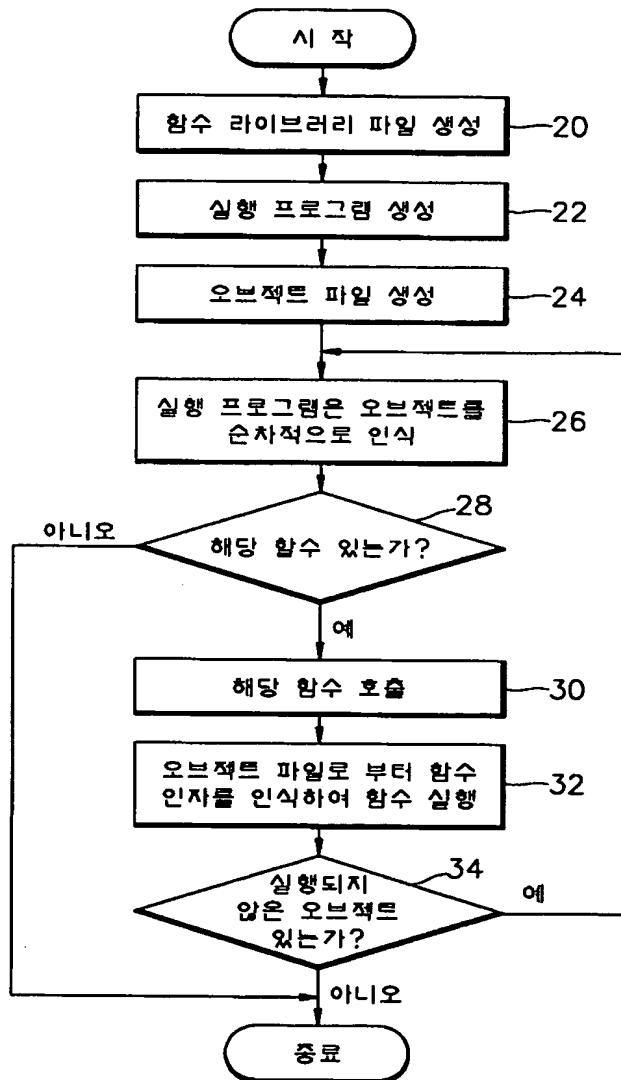
【도 4b】

```
.....  
do while MyArray(1)<>NULL  
    if MyArray(1)="메뉴선택"then  
        Menu(MyArray(1+1))  
    else  
        end if  
.....
```

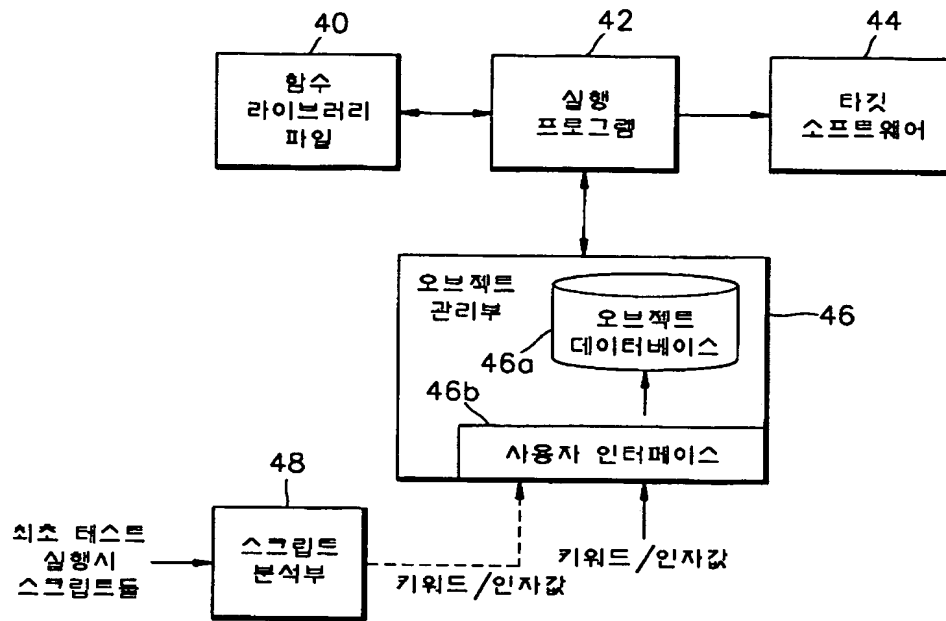
【도 4c】

```
Function Setf(b) <----- 함수시작  
Window SetContext, "Caption="+b+"", ""  
End Function <----- 함수 끝  
  
Function CancelBut(a)  
PushButton Click, "Text="+a+""  
End Function
```

【도 5】



【도 6】



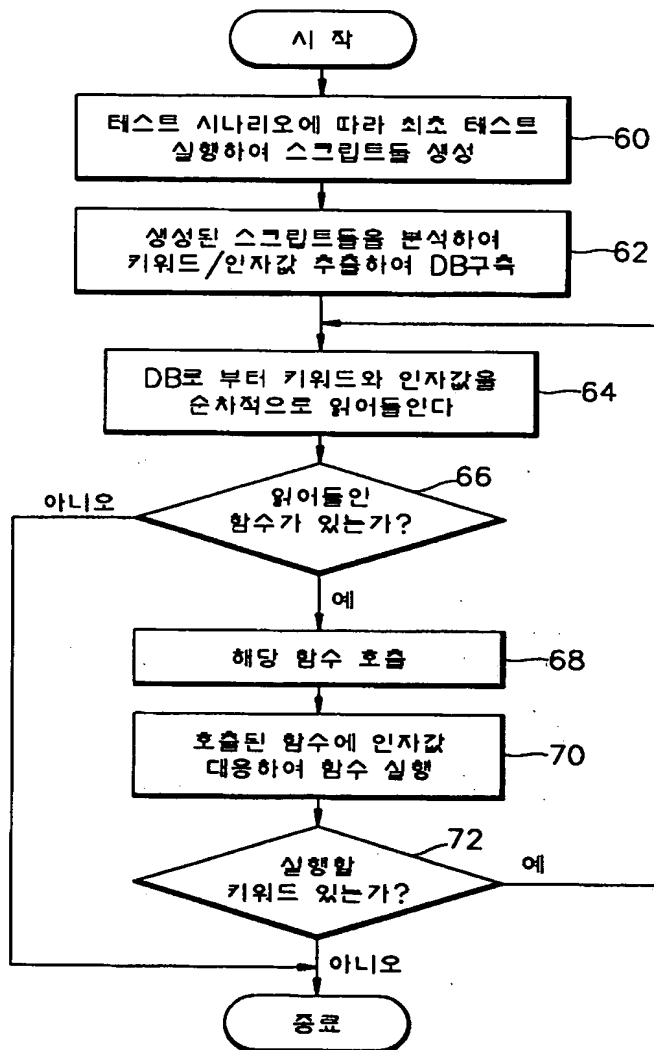
43-35

|      |  |                  |  |
|------|--|------------------|--|
| 키이스1 |  | 메뉴버튼버튼,조회입력,메뉴버튼 |  |
|------|--|------------------|--|

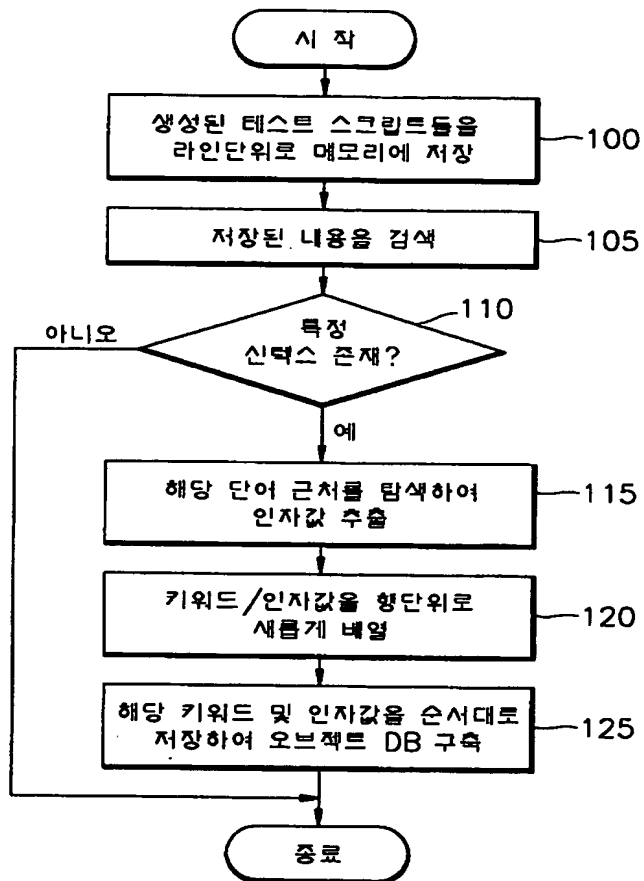
No.  1.00     
 시스템명 : uniERP-2000-VX1     
 부서 : 판매관리

| No. | 구분    | 내역   | 단위                 | 수량  | 비고 |
|-----|-------|------|--------------------|-----|----|
| 1   | 00010 | 기동   |                    |     |    |
| 2   | 00020 | 화면이동 | s211mat            |     |    |
| 3   | 00030 | 조회입력 | s211mat_q1_1       | 1   | 2  |
| 4   | 00040 | 메뉴버튼 | tbQuery            |     |    |
| 5   | 00050 | 버튼   | Server Side Script |     | 복인 |
| 6   | 00060 | 입력   | s211mat_at_1       | 1   | 21 |
| 7   | 00070 | 메뉴버튼 | tbSave             |     |    |
| 8   | 00080 | 버튼   | uniERP(Warning)    |     | 복인 |
| 9   | 00090 | 조회입력 | s211mat_q2_1       | 1   | 2  |
| 10  | 00100 | 메뉴버튼 | tbSave             |     |    |
| 11  | 00110 | 출선택  |                    | 248 | 35 |
| 12  | 00120 | 처리버튼 |                    |     |    |
| 13  | 00130 | 버튼   | uniERP             |     | 복인 |
| 14  | 00140 | 출선택  |                    | 221 | 51 |
| 15  | 00150 | 처리버튼 |                    |     |    |
| 16  | 00160 | 버튼   | uniERP             |     | 복인 |

【도 8】



【도 9】



【H 10a】

```
Sub Main
Dim Result As Integer

'Initially Recorded: 1999-08-26 3:36:04 PM
'Script Name: 00000

Window SetContext, "Caption=Program Manager", ""
ListView DbClick, "ObjectIndex=1;MenuItemText=메모장", "Coords=21,19"

Window SetContext, "Caption=제목 없음 - 메모장", ""
InputKeys "rksksks"
MenuSelect "도움말(H)->메모장 정보(A)"

Window SetContext, "Caption=메모장 정보", ""
PushButton Click, "Text=확인"

Window SetContext, "Caption=제목 없음 - 메모장", ""
Window CloseWin, ""

Window SetContext, "Caption=메모장", ""
PushButton Click, "Text=아니오(N)"

End Sub
```

【표 10b】

| Sub                     | Main    | Dim              | Result              | As          |
|-------------------------|---------|------------------|---------------------|-------------|
| Integer                 | .....선택 | ..      선택 ..... | Window              | SetContext, |
| "Caption=메모장 정보 ,<br>"" | Window  | Window           | CloseWin,           | "" ,        |
| PushButton              | Click   | "Text=아니오 (N)"   | "Caption=메모장 ,<br>" | ""          |



【도 10c】

|     |    |    |    |
|-----|----|----|----|
| 100 | 비율 | 비율 | 비율 |
|     |    |    |    |
|     |    |    |    |
|     |    |    |    |

【도 10d】

|     |         |           |     |
|-----|---------|-----------|-----|
| 100 | 비문<br>근 | 예문장<br>음보 | 아니오 |
| 50  | 단기      | 예문장 정보    |     |
|     |         |           |     |
|     |         |           |     |

【도 10e】

|     |    |        |     |
|-----|----|--------|-----|
| 50  | 타기 | 예문장 정보 |     |
| 100 | 비문 | 예문장    | 아니오 |
|     |    |        |     |
|     |    |        |     |

【표 11】

